Using Reflective Guides to Capture Software Projects Experience

Gerardo Matturro[1], **Andres Silva**[2]
[1]Departamento de Ingeniería de Software, Universidad ORT Uruguay, Montevideo, Uruguay
[2]DLSIIS, Universidad Politécnica de Madrid, Madrid, Spain

Abstract - *Capturing the experience team members acquire in software projects is of great value for organizations wishing to improve their software practices and process. The main drawback of existing methods for capturing this experience is that the capture process is done (if ever) after project completion, which leads to the risk of losing it because, as usually occurs, team members are finally not available to participate. In this paper we present an empirical study of the use of "reflective guides", a knowledge management tool for capturing experience in software projects, which differs from existing approach in the fact that the capturing process is integrated into the daily project activities.*

Keywords: Knowledge management, software engineering, experience capture, reflective guides.

1 Introduction

In software organizations, the knowledge and experience team members acquire in software development projects can be used to improve the development practices in future projects [1]. In this sense, Rus and Lindvall consider that the organizations that are willing to improve the software engineering skills of a team should assure that the knowledge and working experience gained in a project is not lost, and in particular, should identify what went well and what went wrong in relation to the process followed and the software product obtained [2].

For this knowledge and experience to be able to be reused in new projects or in software process improvement initiatives, it first has to be captured, that is, transformed from its tacit form [3] in the mind of team members to an explicit one [3] that enables its dissemination to the rest of the organization.

Common approaches for capturing these knowledge and experiences are based on techniques such as semi-structured interviews ([4], [5]) or by applying methods such as project postmortem analysis ([2]) and post-project reviews ([6]), among others. The main drawback of these approaches is that the capturing process usually takes place at a later time of the occurrence of the experience itself and it is required that the people who own the experience (team members that participated in the project) be available to participate in this capturing process, which in general is something that does not happen. In this paper we introduce the "reflective guides", a knowledge management tool to capture experience, based on the concept of reflective practice applied to the field of software engineering. Our approach to capture software project experience differs from the above mentioned methods mainly in two aspects: the way the experience is captured, and the moment in the project life cycle this capture takes place.

This paper is organized as follows. In section 2 we briefly describe the above mentioned methods, along with a summary of its main criticisms. In section 3 we present the concepts of "reflective practice" and "reflective journals" to put the basement for our distinctive approach. Section 4 is devoted to introduce the reflective guides as a knowledge management tool aimed to enable the capture of knowledge and experience while project unfold. In section 5 we present the case study of the application of our approach in a software organization. Finally, in section 6 conclusion and further works are presented.

2 Existing methods for capturing knowledge and experience

As mentioned above, several strategies and methods have been proposed to enable the capture of knowledge and experiences that members of project teams create and acquire as they carry out a software project. These methods are referred to in literature as project postmortem analysis and post-project revisions, among similar other.

The project post-mortem analysis, as described in [1], comprises the phases: preparation, data collection, and analysis. In the preparation phase all the documentation generated during the project is reviewed in order to understand what has happened, and to determine the goals for the postmortem analysis. The data collection phase is the moment in which the relevant project experience is gathered and, once the important topics have been identified, they are prioritized before proceeding with the analysis phase. During this last phase, a feedback session is conducted in order to analyze the data collected and to find the causes for positive and negative experiences.

The post-project reviews [6] are a way to provide a formal mechanism to transfer experience from a project team to an organizational memory once the project has finished and while these experiences are still fresh in the minds of the participants. The captured experience is stored in a repository of learned

lessons whose purpose is to facilitate the organization, maintenance and spread of the captured knowledge.The repository is based on web technology and it has an interface based on filling-in-forms for the people who provide learned lessons can add new experiences to it.

What these methods (and others alike) have in common is the fact that the process of capturing experience is done later in time with regard to the actual occurrence of the experience, generally after finishing the project or at least by the time it has reached a relevant milestone. Some criticisms arise in relation with this fact. Zedtwitz mentions the restrictions and the lack of time as one of the main reasons for skipping the post-mortem revisions, as organizations usually have a queue of projects that project managers and other team members have to be assigned to, as soon as they are finished with the current one [7]. Cooper mentions the fact that project teams are by definition, temporal entities and that once the project is finished the team members are reassigned to new ones or they return to the organization taking with them their individual knowledge and experience [8]. Oakes considers that once a project is winding down and the team is dispersing, it can be difficult to find energy for such a review and, if the review does happen, it's often little more than an unstructured discussion about people's gripes [9].

3 Reflective practice and reflective journals

Reflection is the practice of periodically stepping back to ponder the meaning to self about what has recently transpired. Reflection illuminates what has been experienced by self, providing a basis for future action [10]. Reflection should be built into every activity, project or work piece in order to maximize learning from everyday activity [11].

But beyond reflection, a further concept of 'reflective practice' has come into greater use. It is defined as "a set of abilities and skills, to indicate the taking of a critical stance, an orientation to problem solving or state of mind" [12]. Schön introduced the reflective practitioner perspective in which professional (architects, musicians and others) rethink and examine their work during and after accomplishing the creative process [13]. According to Schön, two kinds of reflection can be distinguished: reflection-in-action and reflection-on-action. Reflection-in-action takes place as events unfold, where the participant will perceive the situation as new but implicitly compare it to prior experience, situate possibilities for new actions and carry out experiments to decide a course of action. Reflection-on-action happens further away from the event temporally, where the participant will formalize the situation and actions so they can evaluate and think about the situation [13]. In software engineering, an analysis of the field and the kind of work software engineers usually perform, supports applying the reflective practitioner perspective [14].

One traditional tool used for reflection activities is the "reflective journal". A reflective journal records a learning item that took place as a result of reflecting on experiences and situations [11]. In work-based learning settings, journals can be useful in helping participants reflect on experiences, be they in their learning teams, in their projects or just in everyday life [10]. Different uses a learning journal can have are to record experience, to facilitate learning from experience and to enhance reflective practice [15].

4 The reflective guides

Based on the concept of reflective journals presented above, we define the "reflective guides" as a special kind of reflective journal intended to be used by project team members to record the experience they gather during the execution of their software project tasks. A reflective guide includes a series of questions and statements that refer to the software practices, activities, techniques and processes for which knowledge and experience want to be captured, and whose goal is to motivate and facilitate personal reflection activities.

Even though asking questions is nothing new, the difference in our approach with regard to the existing ones described in section 2 is that these questions are given to the team members "before" they perform their project activities and not asked "after" those activities has been performed (as occurs with the methods mentioned above). In this way, respondents know in advance the questions he/she will have to answer later, and find them in a better position to reflect on and to give a more detailed answer. In other words, the idea behind this approach is to avoid the situation in which team members could say "had we known we were going to discuss these topics, we would have paid more attention or gathered some notes at that time". Besides this, having these reflective questions beforehand enables team member to start reflecting and answering them during or immediately after their project tasks are done. This way, by the end of the project, the experience has been captured in the answers to those questions. Here is, precisely, where the main different of our approach resides when compared with the above-mentioned methods: the experience capturing process occurs during the project and not tried to be captured after its end, avoiding the risk of losing that experience because of the criticisms presented at the end of section 2.

To define the types of reflective questions or sentences, we propose to use Bloom's taxonomy of educational objectives [16]. Based on the six levels of this taxonomy, we can formulate questions or sentences that activate different cognitive operations, from the simple recall of facts up to the more complex processes of synthesis and evaluation of information. Three different types of questions can be made for the different levels of Bloom´s Taxonomy. The questions or sentence related to levels 1 (knowledge) and 2 (comprehension) point to knowledge the team members have (or should have), and that should be put into action at the moment they carry out the activities in the project. These types of questions or sentences will motivate the reflection for the action. The questions of levels 3 (application) and level 4 (analysis) must refer to the usage or the practical application of the knowledge or the previous experience that team members

put into practice during the realization of the projects activities. These types of questions should motivate reflection-in-action (in Schön's terminology). The questions of levels 5 (synthesis) and 6 (evaluation) should refer to synthesizing and evaluating the experience lived by team members while doing their project activities. This kind of questions must motivate the reflection-on-action.

For the elaboration of actual reflective questions or statements, the following two elements are taken into account: 1) Concepts related to the software engineering knowledge area, activity, technique or project tasks respect of which it is intended to motivate reflection and to capture experience, and 2) The different levels of Bloom's taxonomy, with their corresponding keywords that reflect the cognitive operations associated with each level. List of keywords can be found in [16].

For instance, if the purpose is to capture experiences related to the "interview" technique for requirements elicitation, with the goal of improving the interaction among software engineers and stakeholders, then some relevant concepts are: "planning the interview", "choosing the interviewee", "types of questions to be asked", "knowledge of interviewee´s terminology", "identification of functional and non functional requirements", etc. Based on these concepts, Table 1 includes examples of reflective questions and sentences for each level of Bloom´s Taxonomy.

5 Empirical study

To study the applicability of the reflective guides, a case study was conducted at ORT Software Factory (hereafter, ORTsf), an academic unit within the Software Engineering department of the University ORT Uruguay. A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident [17].

The conditions of context were particularly important to consider because we aimed to study the use of the reflective guides as elements embedded into the daily working activities of the members of a software project team working in real software development projects, as explained below (sub-section 5.2).

5.1 Research questions

Two research questions were posed for the study: 1) What kind of knowledge and experience can be captured with the reflective guides, 2) How much time respondents invest in answering the reflective guides.

The second research question is motivated by the fact that, as experience indicates, team members usually don't like their activities be interrupted by "additional" tasks, especially for knowledge management, and we wanted to have an indicator of the impact this "additional" activity has in their project-specific activities and in the total time of the projects.

Table 1. Examples of reflective questions

Knowledge area: *Requirements engineering*
Knowledge sub-area: *Requirements elicitation*
Technique**:** *Interview*
Concepts associated: *Choosing the interviewee, planning the interview, knowledge of interviewee's terminology, types of questions to be asked, functional and non functional requirements.*
Questions
Level 1: Knowledge – Cognitive operation: Listing *Make a list of the steps to follow in order to plan the interview.*
Level 2: Comprehension – Cognitive operation: Comparing *Compare the different types of questions that you can make to ask the interviewee.*
Level 3: Application – Cognitive operation: Selecting *What criteria must be taken into account to appropriately select the people to be interviewed?*
Level 4: Analysis – Cognitive operation: Assessing *Assessing the result of the interview, what difficulties and unexpected things do you encounter in your interaction with the interviewee?*
Level 5: Synthesis – Cognitive operation: Modifying *What aspects of planning the interview do you think should be modified for a future instance?*
Level 6: Assessment – Cognitive Operation: Judging *How do you judge the process followed in the interview with regard to the identification of functional and non functional requirements?*

5.2 Selected projects and participants

Three independent software development projects (named COODESOR, GESA, and SCPI) were considered in this study, carried out by 12 students. The project teams were integrated by 3-5 students of the last course of the Systems Engineering career at the University. Each team had a professional support member who acted as a tutor in each project.

Each project had an actual customer, namely, an enterprise or an organization, independent of the University, which needed the software product and to which the products was targeted. These projects were not conceived with the specific purpose of research in itself; they had their own agenda and their own deadlines and objectives, which were agreed beforehand with their respective customers.

Of the twelve students, ten of them have real experience in developing software or in IT related activities in the industry. A working condition for the teams was to work together on-site (in the facilities of the University) for at least 10 hours weekly, in order to promote team cohesion and also to have a similar working ambience to that of a software organization. The remaining 30 expected weekly hours in the project, the students had the freedom to work in the University or in any other alternative place at their choice.

5.3 Reflective guides used in the study

Based on historical data keep at ORTsf regarding the software practices usually evaluated as "deficient" when performed by the project teams, we used the reflective guides to make team members reflect on and capture experience about the processes of defining metrics for software project management.

A reflective guide was elaborated, containing seven questions o sentences regarding this topic. The guides were given to the project manager of each project team. We hold a brief meeting in which we explained the purpose and content of the guides, and how they are supposed to be used as part of their project activities.

5.4 Data collection

The project manager of each team used the guide during their project activities to record his reflections, and returned them back with all the reflective questions answered, along with the time spent in writing those answers.

What follows are extracts of those answers for some of the questions, which will be analyzed in next sub-section. The full guides with the complete answers can be obtained from the first author.

Question 3 corresponds to level 4 (Analysis) of Bloom's taxonomy. Answers are presented in Table 2.

Table 2. Excerpt of the answers to question 3

Question 3: According to your experience, how could you overcome the difficulties for identifying metrics useful for your project?
COODESOR: … *the difficulty we have is identifying metrics that are missing from our point of view…to overcome this I'll be reading the literature on the issue and I'll meet the manager role tutor to learn how we are doing related to metrics.*
GESA: … *we used documentation of previous projects, meetings with role tutors for project management tasks and meetings with the group tutor.*
SCPI: …w*ith the help of the project's tutor, the reviewer and the SQA role tutor… the information available in the ORTsf web site was very useful …*

Question 4 corresponds to level 5 (Evaluation) of Bloom's taxonomy. Answers are presented in Table 3.

Table 3. Excerpt of the answers to question 4

Question 4: What would you recommend to make sure that the selected metrics are appropriate for a given project?
COODESOR: … *always keep in sight the issue of costs, time, etc. … to find useful metrics a manager must keep in mind that it is necessary to see clearly, at first look, the current state of the project at the levels of time, costs, etc.*
GESA: … *I recommend that they look if time is critical to the project or not. From there, some fundamental metrics will come out to give information on the project's development ...*
SCPI: *A first recommendation we learned…is evaluating the effort needed to obtain a given data to elaborate a metric. If the effort is too high or if it requires making several changes in the work process ... it may be better to choose a similar metric, less precise but actual and easy to measure.*

Question 7 corresponds to level 5 (Synthesis) of Bloom's taxonomy. Answers are presented in Table 4.

Table 4. Excerpt of the answers to question 7

Question 7: Of those metrics planning and management activities you carried out in the current project, which ones you consider your performance was adequate and which should be improved?
COODESOR: … *having a talk with the team to make them understand the significance of time records…define the metrics to be used at the very beginning of the project, something that I didn't do because of lack of experience and establishing them after X time after the project had begun it's harder to collect the information needed for metrics to represent reality … the metrics to use have been well selected … avoid collecting metrics that do not contribute too much and that consume more time from the project.*
GESA: *I consider that the activity records of the group members were correctly carried out. I would improve the iteration estimation task …*
SCPI: … *we still cannot say what we did right or wrong… later on, when we use the collected data, we may actually know the errors we made in planning.*

Question 5 corresponds to level 5 (Evaluation) of Bloom's taxonomy. Answers are presented in Table 5.

Table 5. Excerpt of the answers to question 5

Question 5: Report, in at least four or five lines, the lessons learned during the metrics planning process.
COODESOR: …*what I can say is that, given that I work in a maintenance project, I have observed the differences between a maintenance project and a development project.*
GESA: … *metrics used in other projects cannot always be reused, each project must be evaluated by itself and specific metrics defined for that project. What we hear in class or read in books may not always be applicable to the project we are carrying out.*
SCPI: …*we realized that just the "theoretical" framework is not enough, because metrics need to be adapted to the project's reality.*

With regard to the time spent in answering the reflective question, Table 6 shows the times (in hours) devoted by the project managers in answering the reflective guides, along with the times spent in all project management activities and total project times.

Table 6. Times (in hours) devoted to answering the guides and of management activities.

Project	Answering the guides	Proj. Mgmt. activities	Project duration
COODESOR	1.50	95.0	1371.0
GESA	0.82	204.3	1261.7
SCPI	1.17	116.5	2912.0

5.5 Analysis

To answer the first research question, we performed a qualitative analysis of the answers given by the people taking part. This analysis comes from extracting away those elements that are considered to be important or pertinent to answer the research questions and by classifying or grouping these findings with others which might be related [18].

For the first research question, this analysis enabled us to identify sources of knowledge (both implicit and explicit) in the organization, as well as learned lessons and proposals of best practices.

Expressions such as "… meetings with role tutors … and with the group tutor …" (GESA,3), "…with the help of the project tutor, the reviewer and the SQA role tutor…" (SCPI,3) enable identification of tacit knowledge sources. Expressions such as "…we used documentation of previous projects…" (GESA,3), "…the information available in ORTsf web site was very useful…" (SCPI,3) indicate sources of explicit knowledge.

The answers in the guides enable us the identification of lessons learned, derived also from carrying out the project tasks. Expressions such as "…something I did not do because of lack of experience and that were more difficult to collect a long time after the project had begun…" (COODESOR,7), "…if the effort is too high … it is better to choose a similar metric, less precise, but actual and easy to measure…" (SCPI,4), "…metrics need to be adapted to the project's reality…" (SCPI,5) show learned lessons during the project activities.

With regard to the identification of proposals of best practices, expressions such as "…having a team meeting to make them realize the significance of keeping time records…" (COODESOR,7), "…establishing the metrics to be used at the very beginning of the project…" (COODESOR,7) may be considered recommendations to follow that, adequately developed, will allow the formulation of best practices.

To answer the second research question, we proceeded to calculate the percentages of time devoted to answer the reflective guides in relation with the times required to project management activities and total project times (Table 7).

Table 7. Incidence of answering the reflective guides in relation to project times.

Project	Project Mgmt activities	Total project duration
COODESOR	1.58%	0.21%
GESSA	0.40%	0.17%
SCPI	1.00%	0.07%

As we have data of only these three projects, we take these data only as a primary estimation of the effort that implies the use of the reflective guides as part of project activities.

6 Conclusions

In this article we presented an approach for capturing experiences in software projects, by using a knowledge management tool we named "reflective guides".

Different from the pre-existing methods, the proposed approach is based on a previous establishment of the different specific types of knowledge and experiences that are of interest to capture, and on the capture of this knowledge and experience while the project unfold, instead of trying to capture them once the project is over. This way, the problems and inconveniences that arise from postponing the capture of experience after the project ends, as discussed in section 2, are solved.

We have also presented the case study of an implementation of the proposed approach in a software organization. The reflective guide prepared specifically for it was used by the managers of three software development projects to capture their experience about the processes of defining and collecting metrics for software project management. The qualitative analysis of the answers in the guides enabled the identification of three kinds of knowledge artifacts: sources of knowledge in the organization, as well as learned lessons and proposals of best practices. As a quantitative outcome, an indicator of the impact of using these guides in project activities was also obtained.

The study showed that the reflective guides constitute a tool adequate for capturing the knowledge and experience team members acquire during the execution of a software project, and also showed that the use of reflective guides during development does not pose a work overload for the members of the project teams.

Based upon the results obtained in the study, at ORTsf we are planning to formally extend the use of the reflective guides to capture experiences regarding other software engineering process and activities. The starting point for this extension will be those software project activities that, according to our data, are usually performed deficiently by project teams and hence have some potential for improvement, based on the knowledge and experiences captured with the presented approach.

7 References

[1] L. Briand. "On the many ways software engineering can benefit from knowledge engineering". Proc. 14th International Conference on Software Engineering and Knowledge Engineering, pp. 3-6, 2003.

[2] A. Birk, T. Dingsoyr, T. Stalhane. "Postmortem: never leave a project without it". IEEE Software, 19(3), pp. 43-45, 2002.

[3] I. Nonaka, H. Takeuchi. "The Knowledge-Creating Company". Oxford University Press, Oxford, 1995.

[4] S. Komi-Sirvio, A. Mantyniemi, V. Seppanen. "Toward a practical solution for capturing knowledge for software projects". IEEE Software, 19(3), pp. 60-62, 2002.

[5] L. Scott, R. Jeffrey. "An anatomy of an experience repository". Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE 03), pp. 162, 2003.

[6] W. Harrison. "A software engineering lessons learned repository". Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop, 2003.

[7] M. Zedtwitz. "Organizational learning through post-project reviews in R&D". R&D Management, 32(3), pp. 255-268, 2002.

[8] L. Cooper. "Converting project team experience to organizational learning". Proceedings of the 40th Hawaii International Conference on System Sciences, p. 195, 2007.

[9] G. Oakes. "Project Reviews, Assurance and Governance". Gower, Hampshire, 2008.

[10] J. Raelin. "Work-based learning". Prentice Hall, Upper Saddle, New Jersey, 2002.

[11] J. Clifford, S. Thorpe. "Workplace learning and development". Kogan Page, London, 2007.

[12] J. Moon. "Learning journals: a handbook for academics, students and professional development". Kogan Page, London, 1999.

[13] D. Schön. "Educating the reflective practitioner". Jossey-Bass, San Francisco, 1987.

[14] O. Hazzan, J. Tomayko. "Reflection and abstraction in learning software engineering's human aspects"". IEEE Computer, 38(6), 2005, pp. 39-45.

[15] J. Moon. "Learning Journals. A handbook for reflective practice and professional development". New York, Routledge, 2006.

[16] B. Bloom, M. Engelhart, E. Furst, W. Hill, D. Krathwohl, "Taxonomy of educational objectives. Handbook I: Cognitive domain". David McKay, New York, 1956.

[17] R. Yin. "Case study research. Design and methods". Sage, Thousand Oaks, 2003.

[18] L. Blaxter, C. Hughes, M. Tight. "How to research". Open University Press, Philadelphia, 2006.